

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

Warning

This material has been reproduced and communicated to you by or on behalf of *The Charles Darwin University* pursuant to Part VB of the *Copyright Act 1968* (the Act). The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice

Family Name	
Given Names	
Student Number	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>
Teaching Period	Semester 2, 2016

FINAL EXAMINATION	DURATION
HIT220 – Algorithms and Complexity	
	Reading Time: 10 minutes
	Writing Time: 180 minutes

INSTRUCTIONS TO CANDIDATES

EXAM CONDITIONS

You may begin writing from the commencement of the examination session. The reading time indicated above is provided as a guide only.

This is a RESTRICTED OPEN BOOK examination

No calculators are permitted

One A4 sheet of handwritten double-sided notes permitted

Hard copy, unannotated English translation dictionary only

ADDITIONAL AUTHORISED MATERIALS	EXAMINATION MATERIALS TO BE SUPPLIED
No additional printed material is permitted	1 x 20 Page Book 1 x Scrap Paper

**THIS EXAMINATION IS PRINTED
DOUBLE-SIDED.**

**THIS PAGE HAS BEEN INTENTIONALLY LEFT
BLANK.**

QUESTION 1

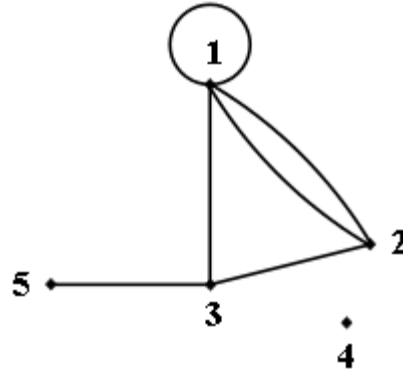
- a) How can it be shown that a decision problem is in the complexity class P? Give an example of a decision problem in P and demonstrate its membership in that class.
(3 marks)
- b) How can it be shown that a decision problem is in the complexity class NP? Give two examples of decisions problem in this class along with brief explanations to demonstrate membership in NP.
(3 marks)
- c) Explain the significance of finding a polynomial time algorithm for a problem in the class NPC. What is Cook's Theorem about?
(4 marks)

QUESTION 2

- a) Calculate the number of times the following loop executes.
- ```
for i := 1 to 2n
 for j := i + 1 to 2n-1
 //do something without branching
 next j
next i
```
- (3 marks)
- b) Assuming that  $n$  is the input size and that “do something without branching” in the innermost loop of part a) involves 100 calculations, state the order of complexity of the entire calculation using big O notation.  
(3 marks)
- c) Another algorithm executes  $2n \log n + 3$  calculations for an input of size  $n$ .
- (i) Is this algorithm  $O(n^2)$ ?
  - (ii) State the complexity of the algorithm using big omega notation.
- (4 marks)

### QUESTION 3

Consider the following graph  $G$ :



- (a) State the adjacency matrix  $A$  for  $G$ .

(2 marks)

- (b) Without calculating  $A^2$ , what information does it record about the graph  $G$ ? State the value of the entry  $(3,2)$  of  $A^2$ .

(3 marks)

- (c) What is the maximum number of edges in a simple graph with  $n$  vertices? Explain the calculation.

(2 marks)

- (d) Is every tree a bipartite graph? Explain.

(3 marks)

## QUESTION 4

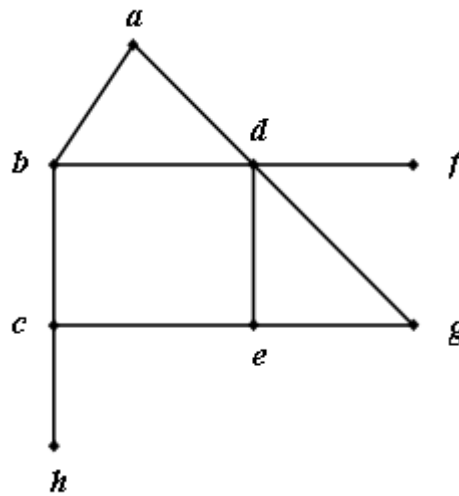
- a) Write pseudocode for an algorithm which uses a stack to match parenthesis. That is, the procedure accepts a string of characters consisting of left and right brackets and determines whether or not the brackets are properly matched.

For example, the string “{ [ ( ) { [ ] } ] ( ) }” has properly matched parenthesis.

(4 marks)

- b) Carry out a **Depth First Search** of the graph below, starting at vertex *a*. Draw a table record the Depth First Index (DFI) as each vertex is visited. Use standard alpha-numeric conventions as appropriate to determine the visitation order. Draw the resultant spanning tree.

(6 marks)



## QUESTION 5

- a) Write code or pseudocode to compute the binomial coefficients according to the following definition:

$$\binom{n}{k} = \begin{cases} 1 & \text{if } k = 0 \text{ or } k = n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{otherwise} \end{cases}$$

(3 marks)

- b) Show a tree of calls for the algorithm in a) for  $n = 4, k = 2$ . Comment on the efficiency of the algorithm.

(4 marks)

- c) Write an iterative algorithm which uses an array to calculate the n-th Fibonacci number.

(3 marks)

## QUESTION 6

- a) Draw a Binary Search Tree that shows the result of the tree after inserting the following keys, in the order listed, into an initially empty tree.

20, 30, 25, 10, 15, 27, 23, 26, 28, 36

(3 marks)

- b) State the result of a preorder and postorder traversal of the BST.

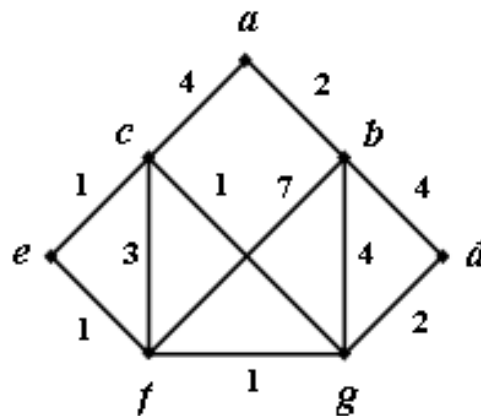
(4 marks)

- c) Use the method of copying to delete the key 25 from the BST, and draw the resultant tree.

(3 marks)

## QUESTION 7

Consider the following graph  $G$ :



- a) Find a minimum-weight spanning tree of  $G$  using **Prims algorithm**, beginning at vertex  $a$ .

Copy the table below into your exam booklet and use it to record appropriate labels at each stage of the algorithm. Draw the resultant tree in your exam booklet. Use standard alpha-numeric conventions, as appropriate, to determine the visitation order.

| Order | $L(a)$ | $L(b)$ | $L(c)$ | $L(d)$ | $L(e)$ | $L(f)$ | $L(g)$ |
|-------|--------|--------|--------|--------|--------|--------|--------|
|       |        |        |        |        |        |        |        |

(6 marks)

- b) Name another graph algorithm that can be used to find a minimum-weight spanning tree of a graph. (1 mark)

- c) Prims algorithm is an example of a greedy algorithm. Name two other greedy graph algorithms and explain how each of them is greedy. (3 marks)



## QUESTION 8

- a) Construct a max heap using the bottom up (Floyd) algorithm on the keys 4,6,3,7,5,4 inserted in that order. Show the construction by drawing the binary tree at each stage that a key is inserted and/or swapped with another key.

(5 marks)

- b) Heap Sort utilises the Floyd algorithm in its first stage to sort an unordered array. Write pseudocode for the second stage of the Heap Sort Algorithm which then transforms the heap into an ordered array. Use ascending order.

(3 marks)

- c) State the worst case complexity of:
- i) Floyd (bottom up) algorithm for heap construction.
  - ii) The entire Heap Sort Algorithm.

(2 marks)

## QUESTION 9

- a) Write a recursive algorithm for computing the number of nodes in a binary tree.  
(3 marks)
- b) Draw the following:
- i) Two different balanced BST's with 5 nodes;
  - ii) A BST with 5 nodes where at least one node has a balance factor of -2.
  - iii) A BST with 5 nodes where each node has a different non-positive balance factor.
- (4 marks)
- c) State the worst case complexity for searching a larger version (with  $n$  nodes) of the BST's drawn in part b) i) and iii).  
(3 marks)

## QUESTION 10

- a) Describe two algorithms which can be classified as “divide and conquer” algorithms. State and explain their worst case complexities.  
(4 marks)
- b) Describe two sorting algorithms which are “in place” algorithms for sorting an unordered array. i.e. The algorithms require little additional storage. Explain the worst case complexities.  
(4 marks)
- c) What is a heuristic algorithm and when is a heuristic algorithm useful?  
(2 marks)